# Проектная работа по модулю SQL.

Анализ данных с сайта LinkedIn

Свитавская Анастасия

# Описание проекта

**Источник данных**

Датасет из > 17 000 вакансий - сайт Kaggle

**Атрибуты**

- **job_id**: идентификатор вакансии, определённый LinkedIn
- **company_id**: идентификатор компании, связанный с публикацией вакансии
- **title**: название должности
- **max_salary**: максимальная зарплата
- **med_salary**: средняя зарплата
- **min_salary**: минимальная зарплата
- **pay_period**: период, за который указана оплата (почасовая, ежемес., годовая)
- **formatted_work_type**: формат занятости (полный рабочий день, неполный рабочий день, контракт)
- **remote_allowed**: разрешена ли удаленная работа
- **formatted_experience_level**: уровень опыта работы (начальный, младший специалист, руководитель и т.д.)
- **type**: вид предоставляемого пособия (401K, Medical Insurance, и т.д.)
- **company_id**: идентификатор компании
- **name**: название компании
- **industry**: сфера деятельности компании

# Постановка задачи

1. Вывод имеющихся IT-вакансий, требования к квалификации/опыту, средняя ЗП
2. ТОП-10 компаний по опубликованным вакансиям
3. Частота предложения бонусов
4. ТОП-10 наиболее оплачиваемых сфер деятельности компаний
5. ТОП-10 востребованных навыков
6. ТОП-10 компаний с наибольшим предложением ЗП
7. ТОП-10 индустрий по опубликованным вакансиям
8. Распределение вакансий по типу занятости

# Предобработка данных в Python

Загрузка данных

в Jupiter notebook

```
import sqlite3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

✓ 11.6s Python

```
# подключение к базе

con = sqlite3.connect('linkedin.db', timeout=10)
cur = con.cursor()
```

✓ 0.0s Python

```
# чтение данных из CSV

benefits_df = pd.read_csv('benefits.csv')
companies_df = pd.read_csv('companies.csv')
company_industries_df = pd.read_csv('company_industries.csv')
company_specialities_df = pd.read_csv('company_specialities.csv')
employee_counts_df = pd.read_csv('employee_counts.csv')
job_industries_df = pd.read_csv('job_industries.csv')
job_postings_df = pd.read_csv('job_postings.csv')
job_skills_df = pd.read_csv('job_skills.csv')
```

Python

# Предобработка данных в Python

## Удаление дубликатов

```
# удаление дубликатов

benefits_df = benefits_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
companies_df = companies_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
company_industries_df = company_industries_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
company_specialities_df = company_specialities_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
employee_counts_df = employee_counts_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
job_industries_df = job_industries_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
job_postings_df = job_postings_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
job_skills_df = job_skills_df.drop_duplicates(keep='first', inplace=False, ignore_index=False)
```

[10] ✓ 0.1s Python

# Предобработка данных в Python

## Удаление лишних столбцов

```
# удаление лишних столбцов

benefits_df = benefits_df.drop(columns = ['inferred'])
employee_counts_df = employee_counts_df.drop(columns = ['follower_count', 'time_recorded'])
companies_df = companies_df.drop(columns=['zip_code', 'url', 'address'])
job_postings_df = job_postings_df.drop(columns=['currency','applies','original_listed_time',
                                                'views','job_posting_url','application_url','application_type',
                                                'expiry','closed_time','listed_time','posting_domain','sponsored',
                                                'work_type','compensation_type'])
job_postings_df = job_postings_df.dropna(subset=['company_id'])
```

[6] ✓ 0.0s Python

# Предобработка данных в Python

## Объединение таблиц

```
[7]  # объединение данных по вакансиям, компаниям, сферам деятельности и численности сотрудников

     companies_df = companies_df.merge(company_industries_df, on='company_id')
     companies_df = companies_df.merge(employee_counts_df, on='company_id')
     job_postings_df = job_postings_df.merge(companies_df, on='company_id')

     ✓ 0.1s                                                                    Python
```

# Предобработка данных в Python

Зарплата в датасете указана по-разному, где-то обозначен диапазон максимальная-минимальная, где-то средняя; отличается и период, за который обозначена сумма (почасовая, месячная, годовая). Приведение ЗП к единому виду:

```python
# приведение ЗП

job_postings_df['normalized_salary'] = job_postings_df.apply(
    lambda x: (x['max_salary']+x['min_salary'])/2 if not pd.isnull(x['max_salary']) else x['med_salary'], axis=1)

period = {'HOURLY': 40*52, 'MONTHLY': 12, 'YEARLY': 1}

job_postings_df['normalized_salary'] = job_postings_df.apply(
    lambda x: period[x['pay_period']] * x['normalized_salary']
    if not pd.isna(x['pay_period']) and not pd.isna(x['normalized_salary'])
    else np.nan,
    axis=1
)
```

✓ 2.8s Python

# Предобработка данных в Python

## Загрузка таблиц в базу данных

```
[11]  ✓ 4.0s                                                    Python

# загрузка данных в БД

benefits_df.to_sql(con=con, name='benefits', index=False, if_exists = 'replace')
company_specialities_df.to_sql(con=con, name='company_specialities', index=False, if_exists = 'replace')
job_postings_df.to_sql(con=con, name='job_postings', index=False, if_exists = 'replace')
job_skills_df.to_sql(con=con, name='job_skills', index=False, if_exists = 'replace')
```

# Анализ данных с помощью SQL

1. Вывод имеющихся IT-вакансий, требования к квалификации/опыту, средняя ЗП.

Для вывода имеющихся вакансий в интересующей сфере создаётся временная таблица, в которую заносится перечень вакансий и ключевых слов для них.

```
CREATE TEMP TABLE spec_names (spec_name TEXT NOT NULL UNIQUE);

INSERT INTO spec_names VALUES
('QA'), ('QE'), ('Quality Assurence'), ('Quality Engineer');

SELECT
    spec_name AS spec_name_or_key_word,
    formatted_experience_level AS exp_level,
    COUNT(title)/2 AS vacancy_amount,
    ROUND(AVG(normalized_salary)) AS midle_salary
FROM spec_names, job_postings jp
WHERE INSTR(lower(title), lower(spec_name)) > 0
GROUP BY spec_name, formatted_experience_level
ORDER BY 3 DESC;
```

| | spec_name_or | exp_level | vaca | midle_salary |
|---|---|---|---|---|
| 1 | QA | Entry level | 6 | 55 467 |
| 2 | QA | Mid-Senior level | 6 | 131 287 |
| 3 | Quality Engineer | [NULL] | 6 | 111 145 |
| 4 | Quality Engineer | Associate | 5 | 70 969 |
| 5 | Quality Engineer | Mid-Senior level | 4 | 150 000 |
| 6 | QA | [NULL] | 2 | [NULL] |
| 7 | Quality Engineer | Entry level | 2 | 87 500 |
| 8 | QA | Director | 0 | [NULL] |

# Анализ данных с помощью SQL

2. ТОП-10 компаний по опубликованным вакансиям

```
SELECT
    name,
    COUNT(job_id) AS vacancy_amount,
    industry
FROM job_postings jp
GROUP BY name
ORDER BY 2 DESC
LIMIT 10
```

| | name | vacancy_amount | industry |
|---|---|---|---|
| 1 | Google | 186 | Computer Software |
| 2 | City Lifestyle | 161 | Publishing |
| 3 | H&R Block | 156 | Retail |
| 4 | Verizon | 113 | Information Technology & Services |
| 5 | Robert Half | 112 | Staffing & Recruiting |
| 6 | Insight Global | 108 | Staffing & Recruiting |
| 7 | Amazon | 93 | Computer Software |
| 8 | The Mom Project | 92 | Internet |
| 9 | Ulta Beauty | 86 | Retail |
| 10 | Milestone Technologies, Inc. | 74 | Information Technology & Services |

# Анализ данных с помощью SQL

3. Частота предложения бонусов

```
SELECT
    b."type" AS benefit,
    COUNT(job_id) AS vacancy_amount
FROM benefits b
GROUP BY 1
ORDER BY 2 DESC
```

| | benefit | vacancy_amount |
|---|---|---|
| 1 | 401(k) | 4 426 |
| 2 | Medical insurance | 2 065 |
| 3 | Vision insurance | 1 989 |
| 4 | Dental insurance | 1 611 |
| 5 | Disability insurance | 1 468 |
| 6 | Tuition assistance | 598 |
| 7 | Commuter benefits | 427 |
| 8 | Paid maternity leave | 417 |
| 9 | Paid paternity leave | 394 |
| 10 | Pension plan | 237 |
| 11 | Student loan assistance | 67 |
| 12 | Child care support | 62 |

# Анализ данных с помощью SQL

4. ТОП-10 наиболее оплачиваемых сфер деятельности компаний

```sql
SELECT
    industry,
    ROUND(AVG(normalized_salary)) AS midle_salary
FROM job_postings jp
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10
```

| | industry | midle_salary |
|---|---|---|
| 1 | Online Media | 295 000 |
| 2 | Veterinary | 275 000 |
| 3 | Investment Banking | 205 000 |
| 4 | Publishing | 195 380 |
| 5 | Public Policy | 185 810 |
| 6 | Computer Software | 148 462 |
| 7 | Consumer Electronics | 144 472 |
| 8 | Fine Art | 144 000 |
| 9 | Executive Office | 141 000 |
| 10 | Pharmaceuticals | 140 505 |

# Анализ данных с помощью SQL

5. ТОП-10 востребованных навыков

```
SELECT
    skill_abr,
    COUNT(job_id) AS vacancy_amount
FROM job_skills js
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10
```

| | skill_abr | vacancy_amount |
|---|---|---|
| 1 | IT | 3 841 |
| 2 | SALE | 2 904 |
| 3 | MGMT | 2 467 |
| 4 | MNFC | 2 195 |
| 5 | BD | 1 993 |
| 6 | ENG | 1 974 |
| 7 | OTHR | 1 574 |
| 8 | HCPR | 1 346 |
| 9 | FIN | 1 227 |
| 10 | ACCT | 813 |

# Анализ данных с помощью SQL

## 6. ТОП-10 компаний с наибольшим предложением ЗП

```
SELECT
    name,
    ROUND(AVG(normalized_salary)) AS midle_salary, industry
FROM job_postings jp
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10
```

| | name | midle_salary | industry |
|---|---|---|---|
| 1 | UP Talent Group | 800 000 | Staffing & Recruiting |
| 2 | ICON Strategic Solutions | 750 000 | Pharmaceuticals |
| 3 | Goliath Partners | 616 667 | Staffing & Recruiting |
| 4 | Selby Jennings | 587 500 | Information Technology & Services |
| 5 | Harris County | 564 213 | Government Administration |
| 6 | Baylor Scott & White Health | 529 338 | Hospital & Health Care |
| 7 | Summit Funding, Inc. | 362 500 | Financial Services |
| 8 | Niantic, Inc. | 317 500 | Computer Software |
| 9 | Acrisure | 317 500 | Financial Services |
| 10 | Rambus | 315 000 | Semiconductors |

# Анализ данных с помощью SQL

## 7. ТОП-10 индустрий по опубликованным вакансиям

```
SELECT
    industry,
    COUNT(job_id) AS vacancy_amount
FROM job_postings jp
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10
```

| | industry | vacancy_amount |
|---|---|---|
| 1 | Staffing & Recruiting | 2 537 |
| 2 | Information Technology & Services | 1 974 |
| 3 | Hospital & Health Care | 1 586 |
| 4 | Retail | 1 235 |
| 5 | Computer Software | 1 007 |
| 6 | Financial Services | 606 |
| 7 | Transportation/Trucking/Railroad | 350 |
| 8 | Construction | 324 |
| 9 | Real Estate | 316 |
| 10 | Insurance | 278 |

# Анализ данных с помощью SQL

## 8. Распределение вакансий по типу занятости

```
SELECT
    formatted_work_type,
    COUNT(job_id) AS vacancy_amount,
    COUNT(remote_allowed) AS remote_vacancy_amount,
    ROUND(COUNT(remote_allowed)*100.00/COUNT(job_id),2) AS percent
FROM
    job_postings jp
GROUP BY 1
ORDER BY 2 DESC
```

| | formatted_work_type | vacancy_amount | remote_vacancy_amount | percent |
|---|---|---|---|---|
| 1 | Full-time | 13 867 | 1 797 | 12,96 |
| 2 | Contract | 1 875 | 570 | 30,4 |
| 3 | Part-time | 1 095 | 85 | 7,76 |
| 4 | Temporary | 136 | 10 | 7,35 |
| 5 | Internship | 118 | 25 | 21,19 |
| 6 | Other | 62 | 7 | 11,29 |
| 7 | Volunteer | 7 | 7 | 100 |

# Выводы

- При востребованности "IT" навыков, в ТОП-10 по оплате данная индустрия не попала
- В топе щедрых работодателей повторяются агентства по персоналу и финансовые услуги
- Рекрутеры же и лидируют по числу открытых вакансий
- Не особо приветствуется удаленный формат работы

# Анализ данных с помощью Python

```
# ТОП 10 компаний по опубликованным вакансиям

job_postings_df['name'].value_counts().head(10)
```

✓ 0.0s Python

```
name
Google                          186
City Lifestyle                  161
H&R Block                       156
Verizon                         113
Robert Half                     112
Insight Global                  108
Amazon                           93
The Mom Project                  92
Ulta Beauty                      86
Milestone Technologies, Inc.     74
Name: count, dtype: int64
```

```
# частота предложения бонусов

benefits_df['type'].value_counts().head(20)
```

✓ 0.0s Python

```
type
401(k)                  4426
Medical insurance       2065
Vision insurance        1989
Dental insurance        1611
Disability insurance    1468
Tuition assistance       598
Commuter benefits        427
Paid maternity leave     417
Paid paternity leave     394
Pension plan             237
Student loan assistance   67
Child care support        62
Name: count, dtype: int64
```

# Анализ данных с помощью Python

```python
# ТОП-10 наиболее оплачиваемых сфер деятельности компаний

job_postings_df.groupby('industry')['normalized_salary'].median().sort_values(ascending=False).head(10)
```

✓ 0.0s Python

```
industry
Online Media                          300000.0
Veterinary                            275000.0
Investment Banking                    205000.0
Publishing                            201182.0
Public Policy                         185810.0
Computer Software                     155575.0
Venture Capital & Private Equity      150000.0
Consumer Electronics                  149250.0
Fine Art                              144000.0
Executive Office                      141000.0
Name: normalized_salary, dtype: float64
```

```python
# ТОП востребованных навыков

job_skills_df.groupby('skill_abr')['job_id'].count().sort_values(ascending=False).head(10)
```

✓ 0.0s Python

```
skill_abr
IT      3841
SALE    2904
MGMT    2467
MNFC    2195
BD      1993
ENG     1974
OTHR    1574
HCPR    1346
FIN     1227
ACCT     813
Name: job_id, dtype: int64
```

# Анализ данных с помощью Python

```python
# ТОП компаний с наибольшим предложением ЗП

job_postings_df.groupby('name')['normalized_salary'].median().sort_values(ascending=False).head(10)
```

✓ 0.0s Python

```
name
UP Talent Group                 800000.0
ICON Strategic Solutions        750000.0
Baylor Scott & White Health     750000.0
Selby Jennings                  650000.0
Harris County                   564213.0
Goliath Partners                425000.0
Summit Funding, Inc.            362500.0
Niantic, Inc.                   317500.0
Acrisure                        317500.0
Rambus                          315000.0
Name: normalized_salary, dtype: float64
```

```python
# ТОП индустрий по опубликованным вакансиям

top_ind_vac = job_postings_df.groupby('industry')['job_id'].count().sort_values(ascending=False).head(10)
top_ind_vac
```

✓ 0.0s Python

```
industry
Staffing & Recruiting                2537
Information Technology & Services    1974
Hospital & Health Care               1586
Retail                               1235
Computer Software                    1007
Financial Services                    606
Transportation/Trucking/Railroad      350
Construction                          324
Real Estate                           316
Insurance                             278
Name: job_id, dtype: int64
```

# Анализ данных с помощью Python

```python
# загрузка данных из БД

job_post = pd.read_sql("""SELECT * FROM job_postings""", con=con)
job_skills = pd.read_sql("""SELECT * FROM job_skills""", con=con)
benefits = pd.read_sql("""SELECT * FROM benefits""", con=con)
company_specialities = pd.read_sql("""SELECT * FROM company_specialities""", con=con)
```

Python

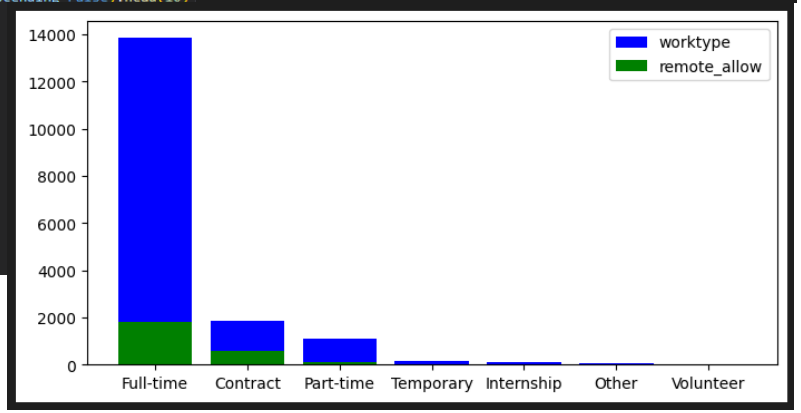
```python
# построение поля для графиков
fig, ax = plt.subplots(figsize=(8,4))

# подсчет вакансий по формату работы и возможности удаленной работы
worktype = job_post.groupby('formatted_work_type')['job_id'].count().sort_values(ascending=False).head(10)
worktype_remote = job_post[job_post['remote_allowed']==1].groupby('formatted_work_type')['job_id'].count().sort_values(ascending=False).head(10)
perc = round(worktype_remote*100/worktype,2)
worktype.name = 'Вакансий всего'
worktype_remote.name = 'Вакансий с удалёнкой'
perc.name = '%'

# настройка графиков
ax1 = plt.bar(worktype.index, worktype.values, color = 'blue')
ax2 = plt.bar(worktype_remote.index, worktype_remote.values, color = 'green')
plt.legend((ax1[0],ax2[0]), ('worktype','remote_allow'))

# объединение подсчетов в одну таблицу
union_WT_RA = pd.concat([worktype, worktype_remote, perc],axis=1)
print(union_WT_RA)
```

| formatted_work_type | Вакансий всего | Вакансий с удалёнкой | % |
|---|---|---|---|
| Full-time | 13867 | 1797 | 12.96 |
| Contract | 1875 | 570 | 30.40 |
| Part-time | 1095 | 85 | 7.76 |
| Temporary | 136 | 10 | 7.35 |
| Internship | 118 | 25 | 21.19 |
| Other | 62 | 7 | 11.29 |
| Volunteer | 7 | 7 | 100.00 |



# Анализ данных с помощью Python

```
# Вывод имеющихся IT-вакансий, требования к квалификации/опыту, средняя ЗП
# 1) создание таблицы для желаемых вакансий

create_query = """
CREATE TEMP TABLE spec_names (spec_name TEXT NOT NULL UNIQUE);
"""
cur.execute(create_query)
```

✓ 0.0s

```
<sqlite3.Cursor at 0x1d3fbfe6e40>
```

```
# 2) заполнение таблицы перечнем желаемых вакансий

full_query = """
INSERT INTO spec_names VALUES
('QA'), ('QE'), ('Quality Assurence'), ('Quality Engineer');
"""

cur.execute(full_query)
# на выбор варианты вакансий и ключ.слов
# ('Java'),('Python'),('Ruby'),('C++'),('C#'),('Kotlin'),('Swift'),('PHP'),
# ('Database Architec'),('Database Engineer'),('Data Engineer'),
# ('Mobile Develper'),('Android Developer'), ('iOS Developer'),
# ('Web Designer'),('JavaScript'),('HTML'),('CSS'),
# ('Network Security Engineer'),('Software Engineer')
```

✓ 0.0s

```
<sqlite3.Cursor at 0x1d3fbfe6e40>
```

```
# 3) запрос искомых вакансий / ключевых слов

select_query = """ SELECT
    spec_name AS spec_name_or_key_word,
    formatted_experience_level AS exp_level,
    COUNT(title)/2 AS vacancy_amount,
    ROUND(AVG(normalized_salary)) AS midle_salary
FROM spec_names, job_postings jp
WHERE INSTR(lower(title), lower(spec_name)) > 0
GROUP BY spec_name, formatted_experience_level
ORDER BY 3 DESC;
"""

it_vac = pd.read_sql(select_query, con=con)
print(it_vac.sort_values(by='exp_level'))
```

✓ 0.4s

| | spec_name_or_key_word | exp_level | vacancy_amount | midle_salary |
|---|---|---|---|---|
| 3 | Quality Engineer | Associate | 5 | 70969.0 |
| 7 | QA | Director | 0 | NaN |
| 0 | QA | Entry level | 6 | 55467.0 |
| 6 | Quality Engineer | Entry level | 2 | 87500.0 |
| 1 | QA | Mid-Senior level | 6 | 131287.0 |
| 4 | Quality Engineer | Mid-Senior level | 4 | 150000.0 |
| 2 | Quality Engineer | None | 6 | 111145.0 |
| 5 | QA | None | 2 | NaN |